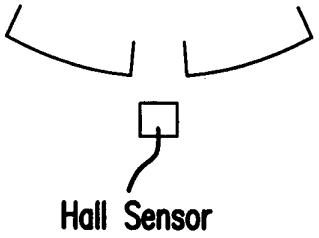
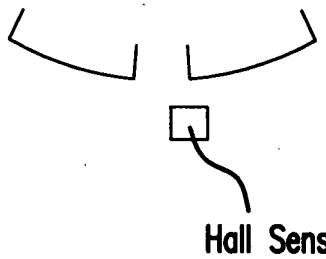
OIPE
DEC 13 2005
PATENT & TRADEMARK OFFICE
1425

PILLSBURY WINTHROP SHAW PITTMAN LLP
Serial No.: 10/803,328 - Confirmation No. 5579
Notice of Allowance Date: November 10,2005
Title: MOTOR SPEED SENSOR....
Attorney Name: Roger R. Wise
Tel.: (213) 488-7100 Docket No.: 63581-308084
Sheet 1 of 8



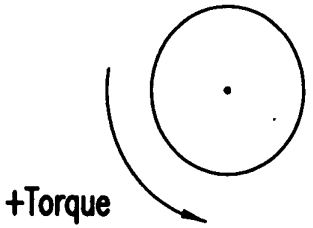
Neutral Position

FIG.1a
PRIOR ART



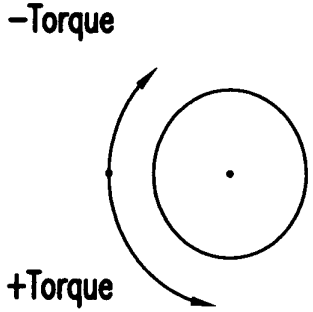
Advanced Position

FIG.1b
PRIOR ART



Neutral Position

FIG.1c
PRIOR ART



Advanced Position

FIG.1d
PRIOR ART

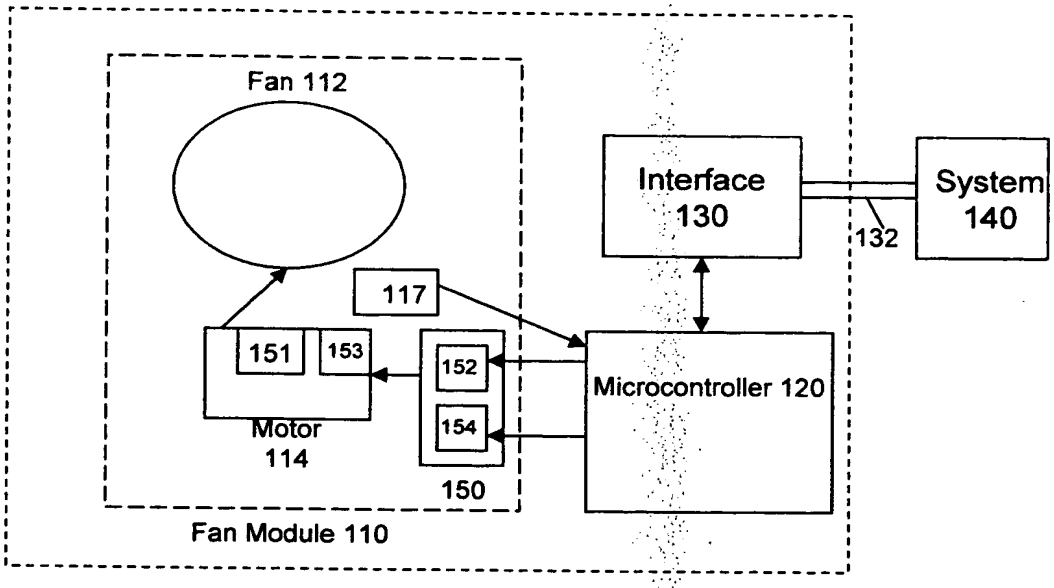

Fan 112
Interface 130
System 140
132
117
151
153
152
154
Microcontroller 120
Motor 114
150
Fan Module 110
Cooling Fan 100

Fig. 2

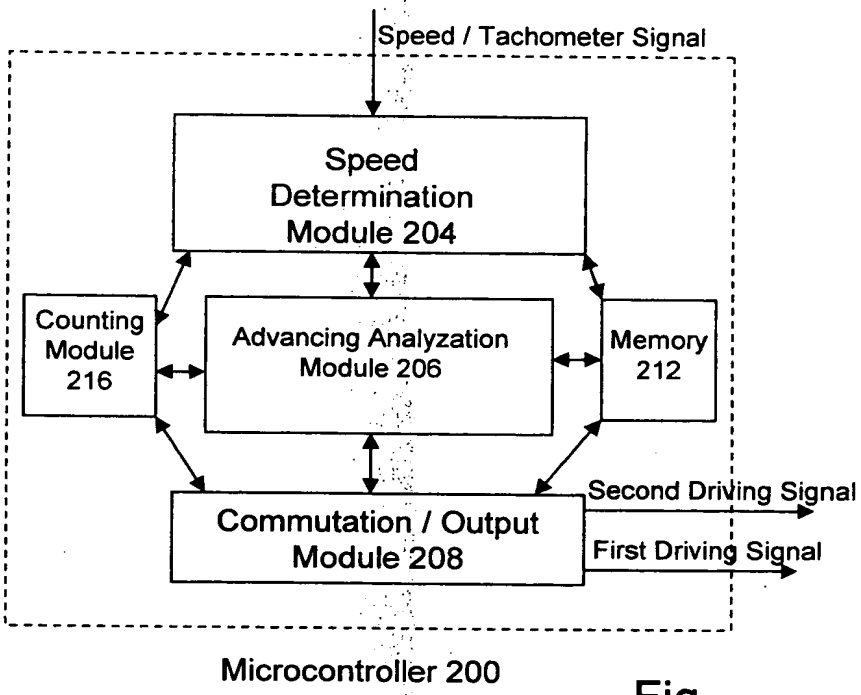

Speed / Tachometer Signal
Speed Determination Module 204
Counting Module 216
Advancing Analyzation Module 206
Memory 212
Commutation / Output Module 208
Second Driving Signal
First Driving Signal
Microcontroller 200

Fig. 3(a)

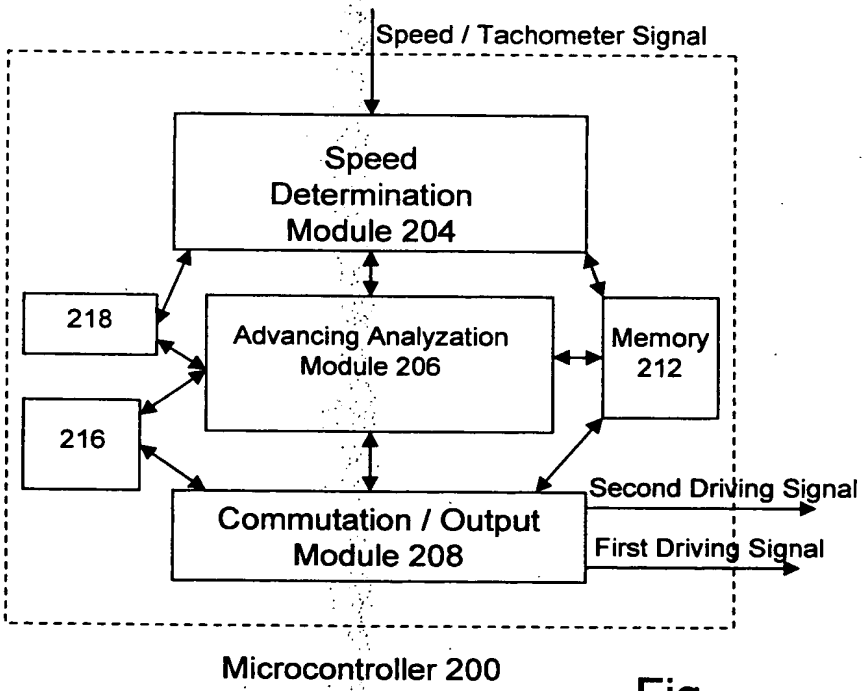

Speed / Tachometer Signal
Speed Determination Module 204
218
Advancing Analyzation Module 206
Memory 212
216
Commutation / Output Module 208
Second Driving Signal
First Driving Signal
Microcontroller 200

Fig. 3(b)

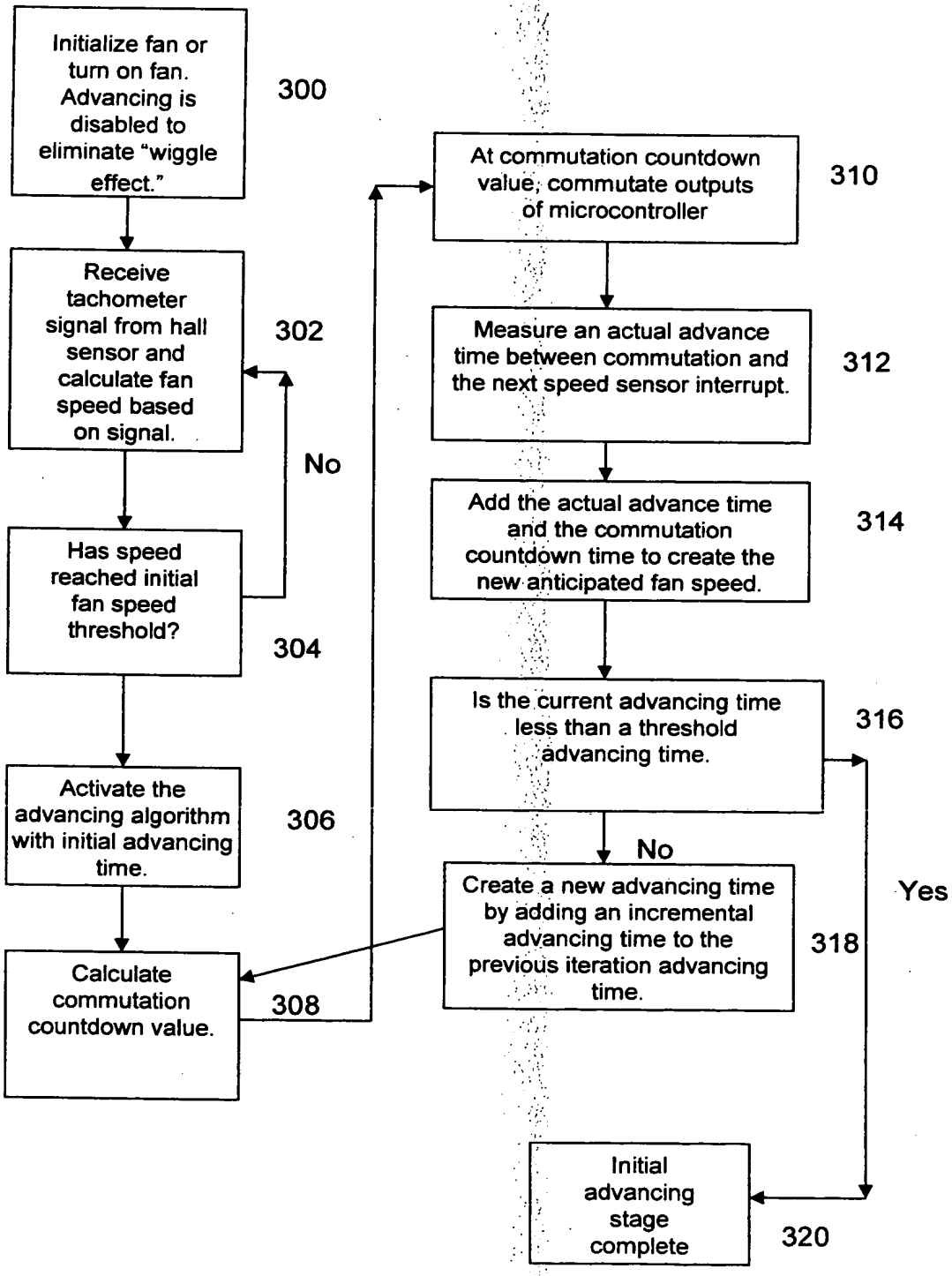

Initialize fan or turn on fan. Advancing is disabled to eliminate "wiggle effect."
300
Receive tachometer signal from hall sensor and calculate fan speed based on signal.
302
Has speed reached initial fan speed threshold?
304
No
Activate the advancing algorithm with initial advancing time.
306
Calculate commutation countdown value.
308
At commutation countdown value, commutate outputs of microcontroller
310
Measure an actual advance time between commutation and the next speed sensor interrupt.
312
Add the actual advance time and the commutation countdown time to create the new anticipated fan speed.
314
Is the current advancing time less than a threshold advancing time.
316
No
Yes
Create a new advancing time by adding an incremental advancing time to the previous iteration advancing time.
318
Initial advancing stage complete
320

Fig. 4(a)

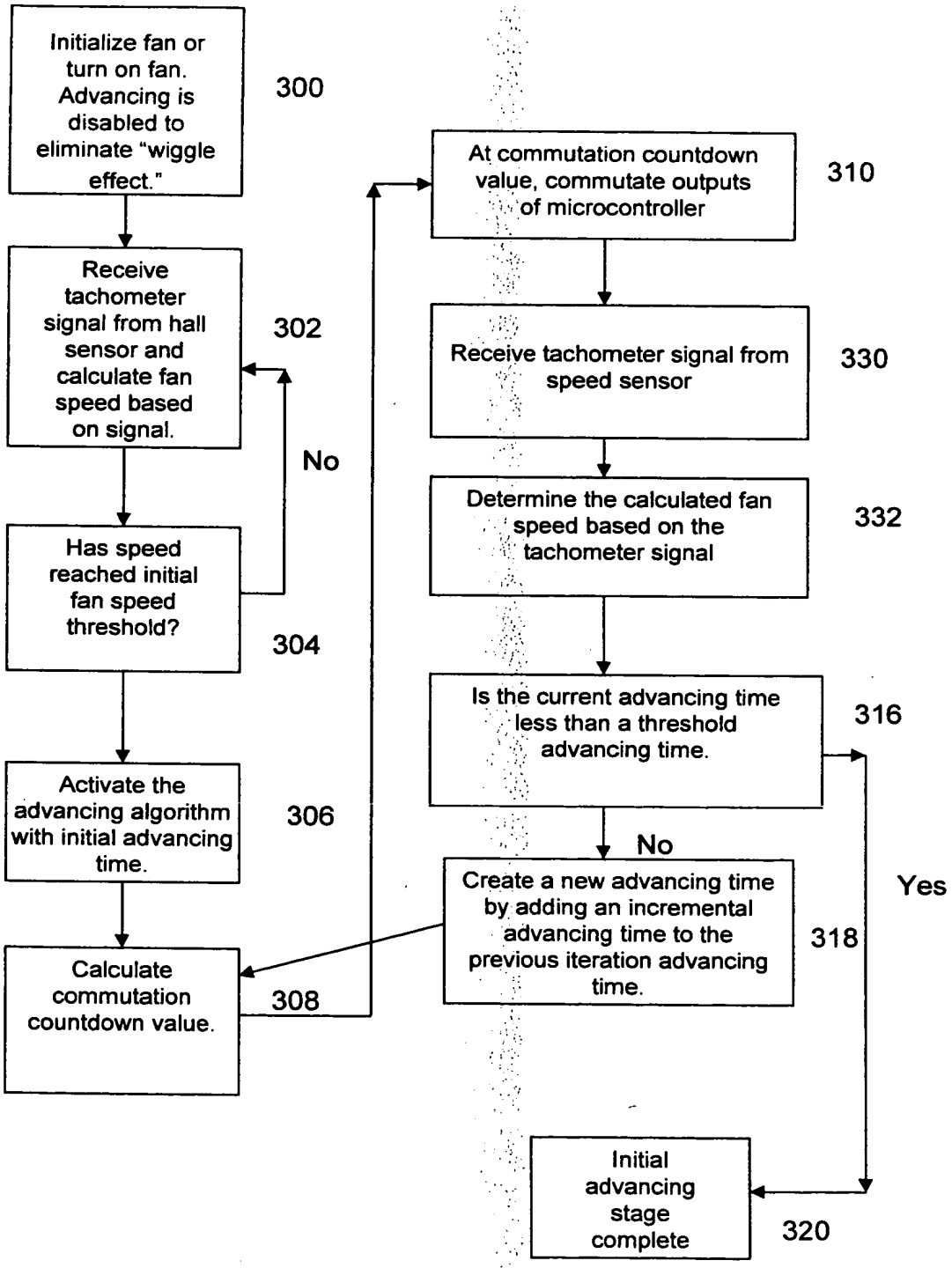

Initialize fan or turn on fan. Advancing is disabled to eliminate "wiggle effect."
300
Receive tachometer signal from hall sensor and calculate fan speed based on signal.
302
No
Has speed reached initial fan speed threshold?
304
Activate the advancing algorithm with initial advancing time.
306
Calculate commutation countdown value.
308
At commutation countdown value, commutate outputs of microcontroller
310
Receive tachometer signal from speed sensor
330
Determine the calculated fan speed based on the tachometer signal
332
Is the current advancing time less than a threshold advancing time.
316
No
Create a new advancing time by adding an incremental advancing time to the previous iteration advancing time.
318
Yes
Initial advancing stage complete
320

Fig. 4(b)

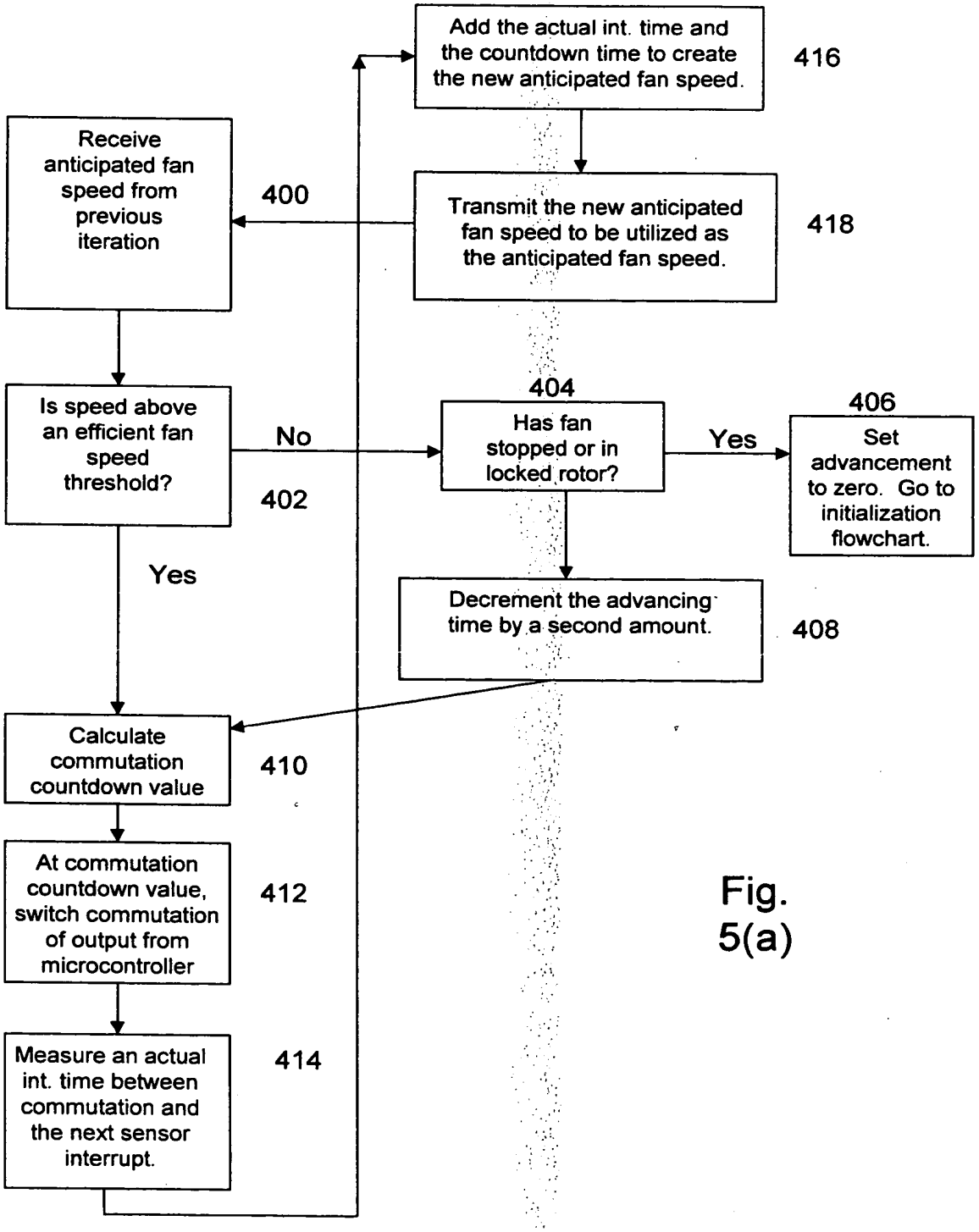

Add the actual int. time and the countdown time to create the new anticipated fan speed.
416
Receive anticipated fan speed from previous iteration
400
Transmit the new anticipated fan speed to be utilized as the anticipated fan speed.
418
Is speed above an efficient fan speed threshold?
No
402
404
Has fan stopped or in locked rotor?
Yes
406
Set advancement to zero. Go to initialization flowchart.
Yes
Decrement the advancing time by a second amount.
408
Calculate commutation countdown value
410
At commutation countdown value, switch commutation of output from microcontroller
412
Measure an actual int. time between commutation and the next sensor interrupt.
414

Fig. 5(a)

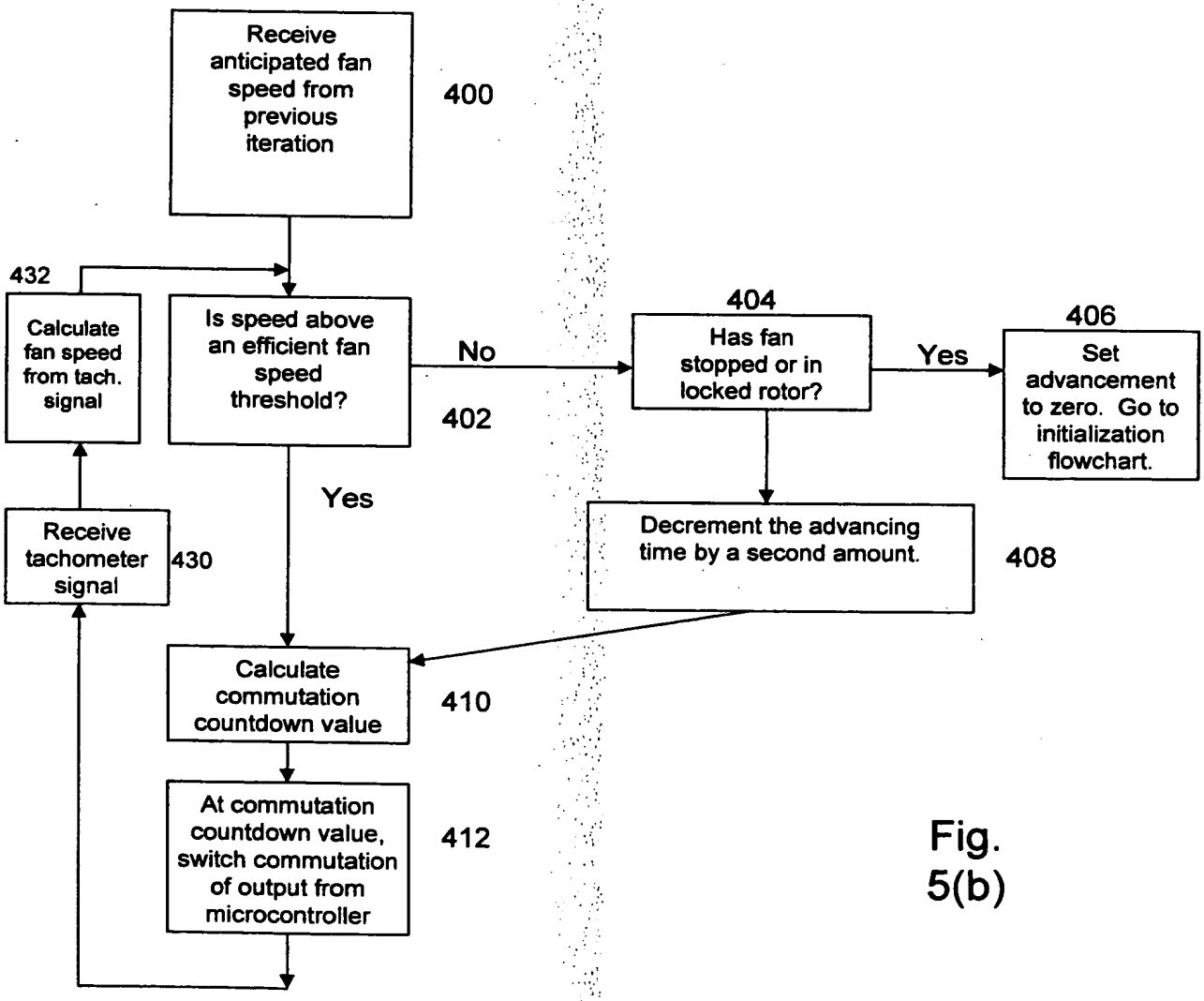

Receive anticipated fan speed from previous iteration
400
432
Calculate fan speed from tach. signal
Is speed above an efficient fan speed threshold?
No
402
404
Has fan stopped or in locked rotor?
Yes
406
Set advancement to zero. Go to initialization flowchart.
Yes
Receive tachometer signal
430
Decrement the advancing time by a second amount.
408
Calculate commutation countdown value
410
At commutation countdown value, switch commutation of output from microcontroller
412

Fig. 5(b)